**04IN1104 Programmiertechniken und Software – Design SoSe 25**

**Table of Contents**

[Software Development Process 2](#_Toc206600975)

[Issues and problems 2](#_Toc206600976)

[Which tasks are not accomplished 2](#_Toc206600977)

[How to configure/ run the software 2](#_Toc206600978)

[What else should be known 2](#_Toc206600979)

[Workload 2](#_Toc206600980)

**Participants**

|  |  |  |
| --- | --- | --- |
| **Name** | **E-Mail** | **Student ID** |
| Sean Schneider | [seanschneider@uni-koblenz.de](mailto:seanschneider@uni-koblenz.de) | 223100290 |
| Mauritz Pötz | [mauritzpoetz@uni-koblenz.de](mailto:mauritzpoetz@uni-koblenz.de) | 220201136 |
| Luca Nolte | [lnolte@uni-koblenz.de](mailto:lnolte@uni-koblenz.de) | 224100156 |
| Silas Heinker | [sheinker@uni-koblenz.de](mailto:sheinker@uni-koblenz.de) | 223200396 |
| Björn Wiesemann | [bwiesemann@uni-koblenz.de](mailto:bwiesemann@uni-koblenz.de) | 223201735 |

**Group Name** Group Foxtrot

# Software Development Process

Our team followed an iterative and agile development process, the process was divided into the following phases, firstly we analyzed the task description provided on OLAT, then we identified the functional and non functional requirements, after that we started with Assignment 10, by designing the UML class diagram and the state machine to “plan” the architecture and get a feeling for the structure. Afterwards we finally started with the implementation, the coding was done in Eclipse. Using Excel we divided the project into smaller tasks with deadlines and held teammeetings via discord twice a week on Wednesdays at 13:30 and Saturdays at 14:30 to review the progreess made and adjust workloads, as well as a few team sessions with our coach Tim to clarify any questions we had and get feedback on our project on Wednesdays at 14:00.

# Issues and problems

Detect and debug errors in the many classes and Java projects   
Coordinate the classes and make everything work  
Properly plan the individual modules and their dependencies

Which tasks are not accomplished  
  
Client Software (CL)

|  |  |  |
| --- | --- | --- |
| **ID** | **Status** | **Note** |
| CL-04 | **Not met** | UI gated only on RDY; no„Wall-Ahead“-check. MazeClientLogic.step() calls protocol.sendStep() directly |
| CL-10 | **Not met** | Dialog only (UI-01). |
| CL-11 | **Not met** | No CLI-Paramaters |

## Source Code (SC)

|  |  |  |
| --- | --- | --- |
| **ID** | **Status** | **Note** |
| SC-03 | **Not met** | Rules not specifically defined |
| SC-07 | **Not met** | SonarQube still says there are some “quality” coding problems |

# How to configure/ run the software

1. Connect to Server: Enter the Maze Server Host and Port in the “Connect” dialog, then press Connect.
2. Log in: Provide a nickname in the login dialog.
3. Strategy: Choose a strategy from the drop-down ( SMART\_BALANCED, SMART\_GEM\_RUSH, SMART\_INTERCEPT\_AGGRO, SMART\_DEFENSIVE, or OFF).

# What else should be known

We ran a one-off SonarQube scan. Several flags are intentional by design (e.g., use of the Singleton pattern per project brief). These were not refactored. A handful of issues are legit and should be fixed. Due to time constraints, they’re left open. Low-impact code smells or purely stylistic suggestions were de-prioritized to avoid churn without user value.

# Workload

We splitted the project as a whole into smaller “subprojects”, as a preperation task we developed the UML diagrams and the statemachine which was done over 3 / 4 days in total including the meeting to talk about the diagrams. Then we agreed on the objectives and planned the whole project which took us 2 days. After that we began with the GameStatusModel which took 6 days to complete, the MazeGameProtocol took 6 days also, then we worked on the CommandHandler which took 5 days, MazeGameClient UI which was done over 12 days, the RobotStrategy took 12 days in total, the Testing as well as the Configurations and the whole final touches took us 3 days. The ProjectReport was done over the span of the whole project and therefore took slighlty over a month.

|  |  |  |
| --- | --- | --- |
| Team member | Tasks | Workload |
| Björn Wiesemann | RobotStrategy, Testing, | 47.5 h |
| Silas Heinker | CommandHandler, Projectreport, Testing | 43.25 h |
| Mauritz Pötz | UML Diagrams, ClientUI, GameStatusModell | 46.5 h |
| Sean Schneider | GameStatusModell, GameProtocol, Client UI | 50 h |
| Luca Nolte | CommandHandler, UML Diagrams | 37.75 h |